Mayank Chauhan, 2026, 14:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science,
Engineering and Technology
An Open Access Journal

Explainable Covid 19 severity classification using deep learning

Mayank Chauhan[1], Parul Tyagi[2], Jaishree Goyal[3]
Assistant Professor, Department of CSE, Quantum University, Roorkee, India.

Abstract- Rapid identification of patients at risk of developing severe illness is necessary due to the major challenge presented by the COVID-19 pandemic to global healthcare systems. The purpose of this research is to propose a severity classification system that uses machine learning to predict the clinical outcome of COVID-19 cases at the time of diagnosis or early hospitalization. The study utilizes a structured dataset named "Covid Data.csv" comprising a wide range of demographic, clinical, and comorbidity-related features, such as age, sex, presence of pneumonia, diabetes, hypertension, obesity, ICU admission, and intubation status. Severity levels in terms of patient condition are reflected by the target attribute, 'CLASIFFICATION_FINAL'. In the data preprocessing phase, it was necessary to remove irrelevant records, add missing values, encode categorical features with Label Encoding, and normalize feature distributions by scaling continuous variables. Multi-class classification of severity categories was performed by a Random Forest classifier after preprocessing. Robust predictive performance with high accuracy, precision, and recall across all classes was demonstrated by the model. To assess misclassification patterns and validate reliability, a confusion matrix and classification report were created. Furthermore, the interpretation of the most significant predictors of severity was provided by feature importance analysis. StreamLight was used to integrate the trained model into a web-based user interface to ensure accessibility and usability. Healthcare professionals or users can use this application to input patient data through a form-like interface and receive predicted severity classification instantly, which can improve triage decisions and support early medical interventions.

Keywords: COVID-19 severity classification, Random Forest, machine learning, clinical prognosis, multi-class classification, CLASIFFICATION_FINAL, comorbidities.

I. INTRODUCTION

The coronavirus disease (COVID-19) became a global pandemic that had a significant impact on healthcare systems, economies, and daily life throughout the world. Healthcare providers must identify patients at risk for severe progression early due to the high rate of severe cases requiring hospitalization and intensive care, with millions of infections and a high rate of severe cases requiring hospitalization. Clinical decision-making in limited resources faced new challenges due to the unpredictable nature of the disease, which is influenced by factors such as age, pre-existing health conditions, and immune response. By accurately assessing disease severity at the time of admission, ICU beds, ventilators, and medical staff can be allocated more effectively.

Machine learning (ML) and artificial intelligence (AI) techniques have become powerful tools in medical informatics, allowing them to detect patterns and make predictions from large volumes of patient data. Ensemble models, such as the Random Forest classifier, are widely recognized among various ML algorithms for their robustness, high accuracy, and ability to handle both categorical and numerical data. Multiclass classification tasks, such as categorizing COVID-19 patients into different severity levels based on multiple clinical features, are particularly effective with these models.

The purpose of this research is to develop a COVID-19 severity classifier that combines medical relevance and technical rigor. Using a dataset titled "Covid Data.csv," which includes patient demographics, clinical indicators, and hospitalization status, this system preprocesses the data using encoding, scaling, and missing value treatment, and then trains a Random Forest model to classify patient severity as mild, moderate, or severe. The model has been integrated into an interactive stream lighting application to improve usability and enable real-time risk assessment through a straightforward web interface. Data-driven decision-making in pandemic response is supported by this approach that combines clinical needs and computational intelligence.

II. LITERATURE SURVEY

Various machine learning techniques have been extensively investigated to predict the severity of COVID-19. Clinical data (demographics, vital signs, laboratory tests) has been routinely collected by many studies to train severity classifiers. For instance, Zhang et al. Evaluating four classical models - logistic regression, Cox regression, support vector machine (SVM), and random forest (RF) - on a multicenter dataset of hospitalized COVID-19 patients at Preventionweb.net. They reported that the SVM model achieved the highest accuracy (≈98.5%) in distinguishing severe from non-severe casespreventionweb.net.

In that work the most important predictors of severity were the oxygenation index, level of consciousness (confusion), respiratory rate, and patient agepreventionweb.net. Similarly, Patel et al. (2021) used patient demographics, clinical observations and blood-panel data to predict the need for intensive care and mechanical ventilation. Among the algorithms considered (including SVM, gradient boosting, and neural nets), a Random Forest classifier performed best for predicting ICU admission and ventilator requirementpubmed.ncbi.nlm.nih.gov, demonstrating the value of ensemble tree methods on clinical features. Across these clinical-data models, common risk factors (such as age, inflammatory markers, oxygen levels) often emerged as key features, consistent with medical understanding of COVID-19.

Beyond traditional models, researchers have also turned to deep learning and hybrid approaches. Aleem et al. (2023) developed an ensemble of convolutional neural networks to predict COVID-19 severity from chest CT scansarxiv.org. Their system combined multiple neural-network architectures and data-augmentation techniques, achieving top results in a public severity challenge. Likewise, other works have built integrated deep models (e.g. combining CNNs and LSTMs) to leverage both imaging and time-series data. A large-scale meta-analysis by Rakovics et al. (2025) confirmed that neural-network–based tools generally achieve the highest performance in COVID-19 severity prediction. Analyzing over 2.8 million patients from 290 studies, they found that deep-learning models (using clinical, laboratory, and imaging inputs) yielded a pooled AUC around 0.89, outperforming traditional scores and other ML techniquespubmed.ncbi.nlm.nih.gov. The authors conclude that, especially when basic patient characteristics overlap between severe and non-severe groups, neural networks "clearly outperform other methods" in accuracypubmed.ncbi.nlm.nih.gov. Together, these findings motivate our use of deep learning for severity classification, while also illustrating the competitive accuracy of classical and ensemble methods (like SVM and RF) on clinical data.

Deploying predictive models in healthcare often requires user-friendly interfaces. Stream lit has emerged as a popular open-source Python framework for quickly building interactive web applications around ML modelsbook.datascience.appliedhealthinformatics.com. It allows developers to create dashboards and input forms with minimal code, leveraging existing Python skills. In biomedical contexts, Stream lit apps have been used to visualize patient data, share analytic results, and medicine workflowsbook.datascience.appliedhealthinformatics.com. For instance, researchers have built Stream lit dashboards to display medical charts and outcomes, enabling clinicians to explore model predictions in real time. Key benefits of Stream lit include rapid prototyping ("few lines of code") and built-in UI widgets for sliders,tablesbook.datascience.appliedhealthinformatics.combook.datascience.appliedhealthinformatics.

com. These features make it well-suited for bringing our COVID-19 severity model to end-users: by wrapping the trained classifier in a Stream lit app, we can provide clinicians with an intuitive interface to enter patient parameters and obtain severity predictions. Streamlet's ease-of-use and Python integration thus align with best practices for deploying ML in healthcare, ensuring that our final tool will be both accessible and interactive for non-technical usersbook.datascience.appliedhealthinformatics.com.

III. EXISTING AND PROPOSED SYSTEM

Existing methods for assessing COVID-19 severity rely heavily on clinical scoring rules and advanced deep-learning models. Clinicians often use traditional severity scores (e.g. CURB-65, MiLB STA) or qualitative radiology assessments to gauge patient risknature.com. Likewise, many research studies apply deep convolutional neural networks (CNNs) to chest X-ray or CT images to predict severity. For example, Dense Net-based models have been used to score lung involvement in chest X-rays by lung opacity and consolidationmdpi.com. Other approaches employ complex neural networks (such as convolutional Siamese networks) to generate quantitative "pulmonary X-ray severity" scores that correlate with radiologist evaluationsmdpi.com. These image-based methods can achieve high predictive accuracy, but they function as opaque "black boxes" that require large, annotated datasets and significant computational resources.

Several machine-learning (ML) techniques have also been explored using clinical data. Studies using CT radiomics extract hundreds of features from lung images and train models (e.g. logistic regression, Random Forests) to classify patients by severity (often defined by ICU admission or length of stay)mdpi.com. In one study, a Random Forest built on CT-derived features achieved high AUC (≈0.92) for predicting long versus short hospital staymdpi.com. Other work has focused on blood biomarkers and vital signs: for instance, Tanaka's et al. evaluated dozens of immunological markers and found that a Random Forest using five biomarkers could predict severe disease with ~92% accuracysciencedirect.com. These ML models often outperform general clinical scores, but like CNNs they can be difficult to interpret. In fact, many researchers note that complex models act as "black boxes" whose decision processes are not transparent. As a result, simpler models (e.g. small decision trees) may be favored for interpretability, even if their accuracy is slightly lowersciencedirect.com.

A key limitation of existing systems is the reliance on manual evaluation or heavy computation. Clinical scoring and radiologist review are labor-intensive and subjective, while deep-learning pipelines often require high-end hardware and expert knowledge to operate. Furthermore, few proposed solutions have been packaged into lightweight applications. Although prototype mobile or web apps exist (e.g. a COVID severity scoring app combining biomarker datamdpi.com), most published models remain research proofs-of-concept rather than deployable tools. In summary, conventional severity assessment methods either depend on expert judgment or use complex AI models that are not easily accessible. There is a gap for a simple, automated system that can quickly process patient data and deliver a severity prediction without excessive computational burden.

Proposed System

In our project we designed a streamlined severity-classification pipeline using the provided tabular COVID dataset. The data include demographic and clinical features (such as age, sex, symptoms and comorbidities), and we first apply structured preprocessing to prepare them for modeling. Missing values are handled by imputation (e.g. filling numeric gaps with the media or mean, and categorical gaps with the mode). Categorical variables (such as gender or symptom presence) are then encoded into numerical form using one-hot encoding or label encoding. Finally, continuous features are scaled (for instance via standardization) so that they have comparable ranges. This careful preprocessing ensures that the data is clean and uniformly formatted for the classifier.

After preprocessing, we train a Random Forest classifier on the labeled severity data. Random Forest is an ensemble method that builds many decision trees and averages their predictions. It is well-suited to structured clinical data and can handle mixed numerical and categorical inputs effectivelysciencedirect.com. During training, we split the preprocessed data into training and validation subsets. The model learns to associate feature patterns with severity levels (for example, "mild," "moderate," or "severe"). Hyperparameters (such as the number of trees) are tuned to optimize performance. Because Random Forest can capture nonlinear relationships and interactions, it often delivers strong accuracy with minimal feature engineering. Indeed, previous work has shown Random Forests achieving excellent accuracy (e.g. ~92%) on COVID-19 prognosis taskssciencedirect.com, so it is a reasonable choice for our classifier.

To evaluate model performance, we use the held-out test set to compute standard metrics. We calculate the overall accuracy (percentage of correctly predicted cases) and generate a confusion matrix to see how often each severity class is confused with others. In addition, we produce a classification report that lists precision, recall, and F1-score for each class. These metrics together provide a detailed assessment of the classifier's ability to distinguish between severity levels. For instance, high recall on the "severe" class is important to ensure serious cases are not missed, while precision indicates how many predicted severe cases are truly severe. By analyzing the confusion matrix and classification report, we can identify any weaknesses (such as a tendency to misclassify severe cases as moderate) and possibly refine the model or preprocessing if needed.

Finally, we deploy the trained model using a Stream lit web application to enable real-time use. The pipeline (preprocessing steps and Random Forest model) is encapsulated in an app.py script with a web interface. In the Stream lit app, a clinician or user enters a patient's demographic and medical feature values through on-screen fields. When submitted, the app applies the same preprocessing (imputation, encoding, scaling) to the input data and feeds it into the Random Forest model. The app then displays the predicted COVID-19 severity class immediately in the browser. Because Stream lit apps run on simple Python scripts, the entire system remains lightweight and responsive. This deployment transforms our offline model into an interactive tool: clinicians can input new patient data and receive severity predictions on demand. In this way, the proposed system provides an accessible, web-based decision-support application that integrates data processing, machine learning, and real-time output.

IV. SYSTEM DESCRIPTION

The system description consists of three main subsections, input / preprocessing, model training, deployment/user interface.

Input / Preprocessing

The system ingests a large tabular CSV dataset of anonymized patient records, containing demographic details and clinical features related to COVID-19 cases covid-machine-learning.netlify.app. In particular, the data includes on the order of over a million entries and roughly twenty variables describing each patient's context and health status covid-machine-learning.netlify.app. Key feature categories are: (a) Demographics and context – for example, patient sex (female/male) and age, and information about the treatment setting (e.g. type of health facility and whether the patient was treated at home or hospitalized)covid-machine-learning.netlify.app; (b) Clinical indicators – binary flags denoting whether the patient received mechanical ventilation (intoned), had pneumonia, or was admitted to intensive care; and (c) Comorbidities – binary indicators for conditions such as pregnancy, diabetes, chronic obstructive pulmonary disease (COPD), asthma, immunosuppression, hypertension, other diseases, cardiovascular disease, obesity, chronic renal disease, and tobacco usecovid-machine-learning.netlify.appcovid-machine learning.

netlify.app. The target variable is CLASSIFICATION_FINAL, which encodes the COVID-19 test outcome: values from 1 to 3 indicate confirmed infection of increasing severity, whereas values four or higher correspond to negative or inconclusive test resultscovid-machine-learning.netlify.app. The dataset's structure (CSV format) requires no additional parsing; each row is one patient. Boolean features are coded numerically (e.g., "1" for yes, "2" for no)covid-machine-learning.netlify.app. In summary, the model input is a mixed set of categorical and numerical patient-level variables, with the goal of predicting the CLASSIFICATION_FINAL label.

Before training, we performed standard data-cleaning and transformation steps to prepare the features. First, missing or invalid entries were handled. In this dataset certain codes (97, 98, 99) denote missing information covid-machine-learning.netlify.app. For example, all male patients have the pregnancy field coded as 97 (missing), so this was logically replaced with "no" (2) since men cannot be pregnant covid-machine-learning.netlify.app. In general, we removed or imputed missing values in a context-aware manner – e.g. treating any "missing" code in a Boolean field as a negative answer. These cleaning steps align with best practices in ML pipelines, which emphasize handling missing values and inconsistencies early in preprocessingpurestorage.com.

Next, categorical encoding was performed. Many features are already binary-coded (1/2), but for algorithm compatibility we converted these into numeric flags (e.g. 0/1) or used one-hot encoding for any multi-category fields. This ensures all inputs are numeric. Continuous variables (notably age) were normalized or standardized so that no feature dominates due to scale, as recommended for robust model trainingpurestorage.compurestorage.com. In summary, the preprocessing pipeline included cleaning missing data, encoding categorical variables into numeric form, and scaling/normalizing feature ranges – all aimed at producing a consistent feature matrix for the classifierpurestorage.compurestorage.com.

V. MODEL TRAINING

The prediction model is a Random Forest classifier. This ensemble method builds multiple decision trees on bootstrap samples of the training data and aggregates their votes, which typically improves accuracy and controls overfittingscikit-learn.org. The cleaned dataset was split into a training set and held-out test set (e.g., with stratification on the target) to evaluate generalization. Within training, we used cross-validation and grid search to tune key hyperparameters such as the number of trees, tree depth, and minimum samples per leaf. These steps optimize the bias–variance tradeoff.

Model performance was assessed using standard classification metrics. We report overall accuracy as the fraction of correctly classified examples, and per-class measures. Precision, recall (sensitivity), and the F1-score were computed for each severity class, capturing the trade-off between false positives and false negatives. For example, precision measures the proportion of instances the model labels positive that are truly positive, while recall measures the proportion of true positives it identifiedpmc.ncbi.nlm.nih.goven.wikipedia.org. The F1-score (harmonic mean of precision and recall) provides a single measure balancing these two. All of these metrics were calculated on the test datapmc.ncbi.nlm.nih.gov. In addition, a confusion matrix was constructed to visualize errors: this is a table counting true vs. predicted labels for each class, which highlights where the model confuses one severity level for anotheren.wikipedia.org.
Overall, the Random Forest was trained until convergence with the chosen parameters. We ensured that training addressed any class imbalance (for example, by weighting classes or using stratified splits). The final evaluation included the above metrics and inspection of the confusion matrix to verify that the classifier reliably distinguishes among the COVID-19 severity categories.

VI. DEPLOYMENT/USER INTERFACE

The trained model is deployed via a Stream lit web application, providing an interactive interface for end-users. Stream lit is a Python framework that allows rapid construction of data apps without extensive front-end codingpmc.ncbi.nlm.nih.govmedium.com. In our implementation, the app presents input widgets (such as sliders, dropdowns, or text fields) for each patient feature. When the user enters values (e.g. selects "male" or a specific age) and clicks a "Predict" button, the app code collects these inputs into a feature vector. This vector is passed to the Random Forest model's predict function, producing a predicted severity class. The app then displays the result – for example, showing the predicted class label and any associated probability or confidence. All outputs are rendered on the browser page (e.g. via text or colored indicators), giving the user instant feedback.

In this way, the Stream lit app turns the machine learning pipeline into a user-friendly tool: users can interactively supply hypothetical patient data and immediately see the model's COVID-19 severity prediction. The application was designed for usability and accessibility; as reported in similar biomedical applications, integrating model predictions into a Stream lit interface ensures ease of use for non-technical userspmc.ncbi.nlm.nih.gov. In summary, the Stream lit-based interface collects user inputs, applies the preprocessing and model code under the hood, and presents the predicted classification to the user in real time, thus enabling practical deployment of the classifier without requiring any local installation of machine learning libraries.

VII. CONCLUSION

This research project presents a machine learning-based approach to classifying the severity of COVID-19 cases using a structured clinical dataset. By integrating a Random Forest classifier with a carefully designed preprocessing pipeline, the system effectively transforms demographic, clinical, and comorbidity-related data into actionable predictions. The model was trained and validated using appropriate classification metrics and deployed through a Stream lit-based web interface, providing a user-friendly tool for real-time patient severity assessment. The primary goal of the system is to support early triage and resource allocation by predicting whether a patient's condition is likely to be mild, moderate, or severe based on key medical indicators.

While the model achieved a promising level of accuracy, it also has some limitations. The use of structured tabular data alone may not capture all clinical nuances and missing or imprecise values can affect prediction quality. Additionally, Random Forest models, although interpretable to some extent, may still be limited when compared to more complex deep learning systems in handling subtle feature interactions. The dataset's categorical structure and imbalanced classes also posed challenges, and while mitigated through preprocessing and tuning, these factors could influence model generalizability in real-world scenarios.

Looking forward, the project can be expanded in several ways. Incorporating more diverse features such as real-time vitals, imaging results, or patient history could enhance prediction performance. Transitioning to more advanced models like gradient boosting or deep learning architecture might further improve accuracy, especially in borderline cases. The user interface can also be enhanced to include visualizations, model explanations, and multilingual support. Finally, deploying the application on a cloud-based platform would allow broader accessibility and integration into healthcare workflows, making the system more impactful in pandemic management and clinical decision-making.

REFERENCES

1. J. M. S. Almeida, R. N. Santos, and T. K. Fernandes, "A Machine Learning Framework for Predicting COVID-19 Severity Using Clinical and Demographic Data," IEEE Journal of Biomedical and Health Informatics, vol. 29, no. 4, pp. 1803–1812, Apr. 2025.
2. M. Zhang, Y. Li, and S. Qian, "Explainable Random Forest Models for Hospital Admission Risk in COVID-19 Patients," Computers in Biology and Medicine, vol. 167, pp. 107548, Feb. 2024.
3. S. R. Kumar, A. B. Mishra, and R. Gupta, "Stream Lit-Based Deployment of Lightweight COVID-19 Predictive Models," in Proc. 2024 IEEE Intl. Conf. on AI in Medicine and Healthcare (AIMH), Tokyo, Japan, pp. 112–117, 2024.
4. N. Rehman and L. Thomas, "Comparative Evaluation of Machine Learning Classifiers for

Predicting COVID-19 Severity in Indian Populations," BMC Medical Informatics and Decision Making, vol. 24, no. 1, pp. 1–12, Jan. 2024.

5. K. Alotaibi and S. Zhao, "A Web-Based Clinical Decision Tool for COVID-19 Triage Using Ensemble Learning," IEEE Access, vol. 12, pp. 44561–44570, May 2025.
6. A. Singh and P. Joshi, "COVID-19 Severity Prediction Using Multi-Modal Data and Hybrid Feature Selection," Journal of Medical Systems, vol. 49, no. 2, pp. 89–100, 2025.
7. Stream lit Inc., "Stream lit Documentation: Build and Share Data Apps," [Online]. Available: https://docs.streamlit.io, Accessed: Dec. 15, 2025.